DOCUMENT RESUME

ED 030 451 LI 001 486

By-Reilly, Kevin D.

Digital Computer Simulation Models of Library-Based Information Retrieval Systems. Reports on File Organization Studies.

California Univ., Los Angeles. Inst. of Library Research.
Spons Agency-National Science Foundation, Washington, D.C.
Pub Date [Jan 69]
Note-34p.; Paper delivered at the 3rd Annual Computer Science and Statistics Symposium, Los Angeles, Calif., January 30-31, 1969.
EDRS Price MF-$0.25 HC-$1.80
Descriptors-Automation, *Computer Programs, Computers, *Information Retrieval, Information Storage, *Information Systems, Library Materials, Library Services, *Models, *Simulation, Use Studies

A simulation study of library-based information retrieval systems is described. Basic models for each of several important aspects are presented: (1) user behavior, emphasizing response to quality and delays in services; (2) the scheduling of services and the organization of the machine-readable files; and (3) the distribution of conventional library materials. Many of the variables in the model (e.g., user expectations of service time, delays involved in providing services, etc.) are random. An adaptive element is present in the form of admitting changes in user expectations in response to system changes. The need for frequent model change and the merging of component models into a comprehensive model for the entire system is established as the basis for use of a user-oriented simulation language. Statistical facility beyond that supplied in the language used (GPSS/360) is required to achieve the goals of this and related investigations. (Author)

EDO 30451

U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
OFFICE OF EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL OFFICE OF EDUCATION POSITION OR POLICY.

47 001486

DIGITAL COMPUTER SIMULATION MODELS OF LIBRARY-BASED INFORMATION RETRIEVAL SYSTEMS

By

Kevin D. Reilly

One in a Series of Reports on
File Organization Studies
(NSF Grant GN-422)

A Paper Delivered at

The 3rd Annual Symposium
Computer Science and Statistics Symposium
Los Angeles, California
January 30-31, 1969

AIC/CLIS
JUN 16 '69
UNIVERSITY OF MINNESOTA

Institute of Library Research
University of California
Los Angeles, California

47 001486

DIGITAL COMPUTER SIMULATION MODELS OF LIBRARY-BASED INFORMATION RETRIEVAL SYSTEMS

By

Kevin D. Reilly

One in a Series of Reports on
File Organization Studies
(NSF Grant GN-422)

A Paper Delivered at

The 3rd Annual Symposium
Computer Science and Statistics Symposium
Los Angeles, California
January 30-31, 1969

Institute of Library Research
University of California
Los Angeles, California

TABLE OF CONTENTS

ABSTRACT i

AN OVERVIEW 1

Previous Studies

Aspects for Modelling

COMPUTER PROCESSING CENTER MODEL 5

Implementation

Extensions

ORDER-AND-DELIVERY SYSTEM MODEL 13

USER-BEHAVIOR MODEL. 15

Formulation of the Model

Elementary Model Results

Evaluation Procedures

GPSS AND STATISTICS. 24

REFERENCES 30

ABSTRACT

A simulation study of library-based information retrieval systems is described. Basic models for each of several important aspects are presented: user behavior, emphasizing response to quality and delays in services; the scheduling of services and the organization of the machine-readable files; the distribution of conventional library materials Many of the variables in the model (e.g., user expectations of service time, delays involved in providing service, etc.) are random. An adaptive element is present in the form of admitting changes in user expectations in response to system changes. The need for frequent model change and the merging of component models into a comprehensive model for the entire system is established as the basis for use of a user-oriented simulation language. Statistical facility beyond that supplied in the language used (GPSS/360) is required leading to an inquiry into the ingredients of a supplementary program package to achieve the goals of this and related investigations.

AN OVERVIEW

This discussion is intended to provide an overview of some digital simulation efforts in a very large area of study roughly characterized by the terms: information retrieval networks or library-based information retrieval systems.

Previous Studies

A word about the context in which the present study is found is useful. In the early days of modelling of information storage and retrieval systems, researchers dealt primarily with cost and timing factors for the automated subsystem. Studies of this type include those of Bourne and Ford (1964) and Blunt et. al. (1966). Concern has spread to other aspects of the study: the role of the user in the information process (Reilly and Hayes, 1967); the environment in which the automated system is found including the interface with non-automated systems (the library for us) and the role of policy and funding decisions (Baker and Nance, 1968). Most of these efforts have at least toyed with the idea of a "general" model applicable to a reasonably wide-range of systems of a certain type. We shall see, however, that difficulties have arisen in this attempt, when we discuss one of these "general" models later. The task of achieving generality becomes more acute when the scope of the model becomes larger. Generality in the sense of macro components seems to be the only hope here; we shall meet with this concept again later.

Aspects for Modelling

The additions that we must make in order to account for features beyond the automated systems extend the scope of the entire project to include the following:

users: people, seeking information;

a computer-processing center (or centers);

storehouses of conventional library materials (books, etc.)

When it comes to users we may think of user-computer dialog--the on-line systems so much discussed today. Alternatively, we may think of the user involved in the business of information gathering: creating demands for services and experiencing alternately frustration and satisfaction when services are not or are provided.

When it comes to computers and information retrieval we may think of file management, e.g., strategies for file organization, the maintenance and updating of the file, and the kinds of retrieval services to be provided. Also we may think of problems of scheduling of services: the query input rate and the flow of queries through the system.

When it comes to storehouses of conventional materials we may think of now-prosaic self-inquisition, the favorite among information specialists: "Where are we going to put it all?" We may think of the libraries' need to adopt microreproduction for space-saving and widespread storage of materials of book length, journals, and report literature. Alternatively, we may imagine a large central library with comprehensive holdings in all forms and connected by facsimile and microwave transmission to local libraries. (Knowledge of problems in this area and the proposed solutions may be expected to be fairly widespread since specialists outside of the library and information science community have put forth their views (e.g., Kemeny, 1962; Brown, et. al., 1967).)

We have developed models in each of the three areas just mentioned. We have not yet incorporated every aspect in each of these. We have, however, included most of the features that Baker and Nance (loc. cit.) reported as lacking in previous studies. The models have already been used in discussions and for suggesting experiments in connection with a state library network in California (Reilly, 1968a) and with a medical library network (to be published). A full-scale experimental and simulation project for the latter area is envisioned for the future.

Our objective from the beginning has been to be able to use the system of component models in an integrated fashion. A couple of difficulties must be overcome in so doing. The first and most obvious one is that the extensiveness of the effort is almost prohibitive; the experimental work by Goodman et. al. (1966) and Goodman and Jones (1968) on information-seeking habits of scientists and technicians in industry and the work by Blunt et. al. (1966) on the simulation of the (subsystem) information retrieval processing center attest to the truth of this statement. Even with the (strictly) subsystem models we must compromise on many matters: we must classify individual material items; we must group users; we must restrict our attention to only part of the "information process" (Reilly, 1968b). The second difficulty is that we must contend with disparate event-time scales for each of the portions of the systems, as illustrated in Figure 1.

EVENTS	TIME RANGE
computer processing	nanoseconds to milliseconds
user decisions	seconds to minutes
ordering and delivery of materials	minutes (several) to days

Figure 1: Illustration of time scales relevant to activities in an information retrieval network.

Part of the solution to the first of the difficulties has already been mentioned. The rest of the solution to that difficulty and to the second difficulty lies in the development of a hierarchy of models similar in concept to that mentioned by Jones (1968). The results from running a model at one (smaller-time scale) level are written into disk data sets and made available to the next higher level model. The necessity of working with the small-time events first is apparent. The data developed in the lower-level models are in the form of histograms and system-monitoring statistics (numbers) for decision processes that transactions moving through the higher-level model blocks may utilize. The models are designed to be run back-to-back with no programmer intervention, if so desired.

COMPUTER PROCESSING CENTER MODEL

A discussion of the full effort can begin with the model of Blunt (loc. cit.), the main focus of which is scheduling within the computer processing center (see Figure 2). Some terms that can be applied to this model, which indicate its scope, are the following:

multiple entry points for queries (possibly remote);

multiple entry modes (telephone, courier, etc.);

ditto for the exiting of responses

activities such as card punching, card-to-tape conversions, card/tape transporting;

macro-level processing times (e.g., high and low levels of file search, editing, output, etc.);

error detection and correction (rewriting);

availability times for input and system facilities.

Implementation

We have programmed a model of this type. The program, in GPSS consists of two separate jobs, an initial job which writes a magnetic tape (a so-called JOBTAPE) and a follow-up job that processes the tape.

Basically, three types of data are assigned to parameters of transactions (the queries) in the first program: identification information; delay times at various service units; numbers for logical and probabilistic routing of transactions through the service units. (Other parameters, called auxiliary parameters, are used in the tape writing programs; but, to a large extent, their use constitutes a program facilitation detail that we need not discuss here.) The information that goes into the parameters

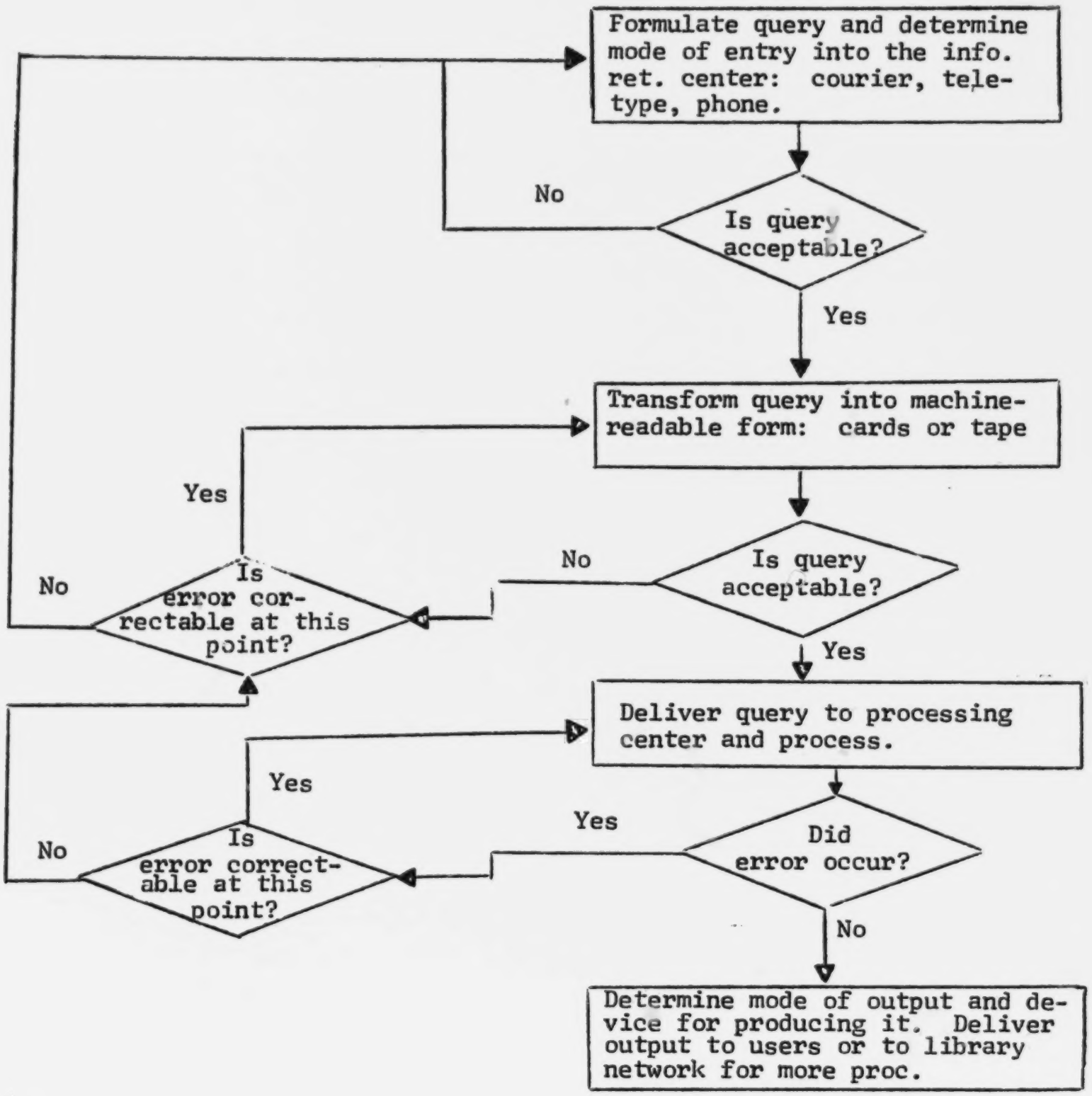

Formulate query and determine mode of entry into the info. ret. center: courier, teletype, phone.
No
Is query acceptable?
Yes
Transform query into machine-readable form: cards or tape
Yes
No
Is error correctable at this point?
No
Is query acceptable?
Yes
Deliver query to processing center and process.
Yes
No
Is error correctable at this point?
Yes
Did error occur?
No
Determine mode of output and device for producing it. Deliver output to users or to library network for more proc.

Figure 2: Gross level chart of information retrieval center processing events. Model adapted from C. Blunt et. al., 1966.

is derived by FUNCTIONS and VARIABLES from the query code (or generalizations thereof, our auxiliary parameters), illustrated in Figure 3. Function numbers for delays, probabilities for errors and routing, etc. are assigned from parsing the query code (e.g., a "1" in position number 2 of the code indicates the need to use certain "low" (shorter) time functions for delays in searching, sorting, editing, etc.).

The program itself has two "control" loops (one for timing and the other to restrict query production to certain times of the week). The only limit on the number of days to be simulated in any one run is the user's budget; the number of days simulated is controlled entirely by a single control card. The other "control" loop turns logic switches on and off according to the schedule desired; the switches control whether or not queries that are generated are allowed to enter the system. The "guts" or main portion of this program contains the parameter assignment statements. These statements are situated after the GENERATE blocks (we have one generate block for each query type) for assignments that are unique to each query type. Another section of code that is branched to by all types of queries provides a list of assignment statements shared by all the different kinds of queries.

The second program reads the JOBTAPE provided by the first. The JOBTAPE contains the relative time at which each transaction is to enter the system. After entering, each transaction follows its allowed path through the system; this path may be tortuous since branching at certain points is probabilistic. Figure 4 illustrates the kinds of delays, error processing, etc. experienced by query-type-1101 transactions. Most of the rectangular blocks consist of a series of statistics gathering blocks, facility or storage entities, and one or more TRANSFER blocks. (Basically, three separate strings of code could model almost all of these

QUERY CODE		MEANING		
	SITE	METHOD OF INQUIRY	SEARCH	OUTPUT COMMUNICATIONS
1101	A	Telephone	Low	Courier/Data Link
1102	A	Courier	Low	Courier
1201	A	Telephone	High	Courier/Data Link
1202	A	Courier	High	Courier
2201	B	Telephone	High	Courier/Data Link
2202	B	Courier	High	Courier
2303	B	Teletype	Lo-Hi	Data Link
3101	C	Telephone	Low	Courier
3102	C	Courier	Low	Courier
3103	C	Teletype	Lo-Hi	Data Link

Figure 3: Query types, from paper by Blunt, et. al. (1966).

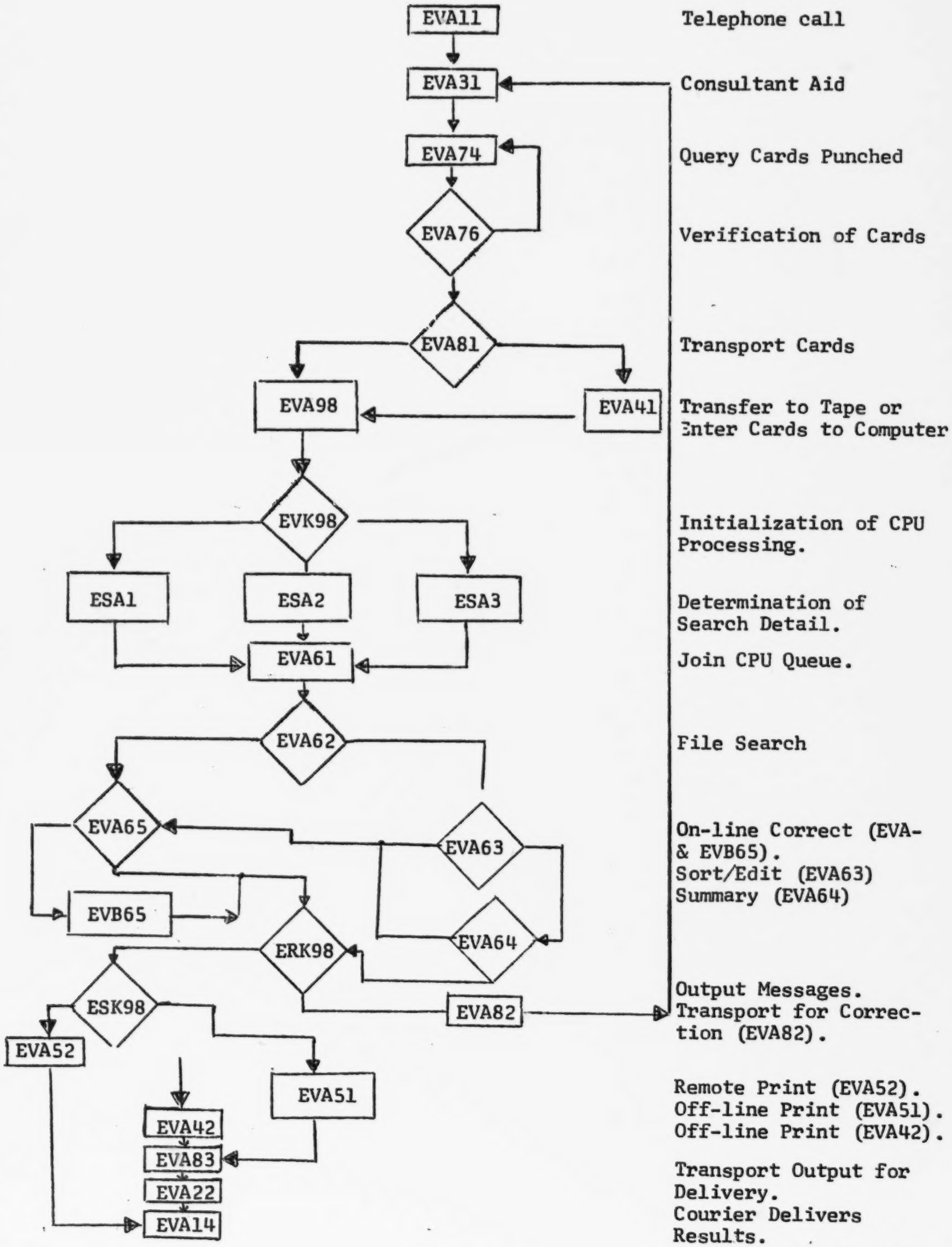


Figure 4: Pathway of Query Type 1101 through the system.

blocks; this implies that three basic macroes could be used to write the entire system description.) The triangularly-shaped blocks are the transfer blocks; it can be seen that they provide for alternate pathways and for the possibility of rerouting (for error correction). Once more "control" loops (for time and for facility, etc., availability) exist. Another loop exists for unscheduled breakdowns, represented generally as a facility-preempt states.

The output from this model develops in snaps taken at various intervals. So far we have utilized the normal GPSS output:

FACILITIES

average utilization
number of entries
average time of transit
seizing transaction

STORAGES

maximum contents
average contents
capacity
average utilization
number of entries
average time of transit
current contents

QUEUES

maximum contents
average contents
total entries
zero entries
percent zeros
average time of transit
" " " "
table number
current contents

It is simple to detect where bottlenecks occur in the system with this kind of output. We have run a model similar to Blunt's example model and have achieved somewhat similar results. Some effort is being put into the task of making the two models achieve almost identical results. Simple

alternative models have been tried out (e.g., additional availability time for the CPU). Addition of cost aspects of the model is being investigated by a student. These changes and additions could well amount to be substantial. They are not, however, the full range of alterations (see below). Reports describing this effort and the results from simulating the alternative cases will be available in the not too distant future.

Extensions

A simple reprogramming of a model like the Blunt model in GPSS strengthens it some: statistic-gathering blocks can be added most easily; restrictions on generality of certain variables and functions in the older model are readily cast aside, etc. More substantial additions consist of admission of a larger number of sites, and the facilities, etc. for handling them, preparing for further types of input/output (e.g., on-line query) facilities, etc. (Some of these changes can be "grafted" to the second (system) model without changing the first JOBTAPE-writing program.)

A substantial change of critical importance to us is merging it with another model for a computer-processing center that has also been implemented in a simple form (see Figure 5 for a gross-level flow chart of this second model). This second model turns toward a more automatic scheduling system as opposed to the external rigidly-imposed scheduling system in the Blunt model. The new (merged) model will have a number of new features:

update input as well as query input.

a priority system for update and queries depending on factors such as importance and size (updates); professional level, site of input, etc. (queries).

automatic reallocation of priorities by the system.

output of messages into the ordering-and-delivery system.

The merging of these two models represents a major test of our assumption that programming-change ease in GPSS offsets any difficulties we may encounter in using GPSS, e.g., in programming more advanced statistics.

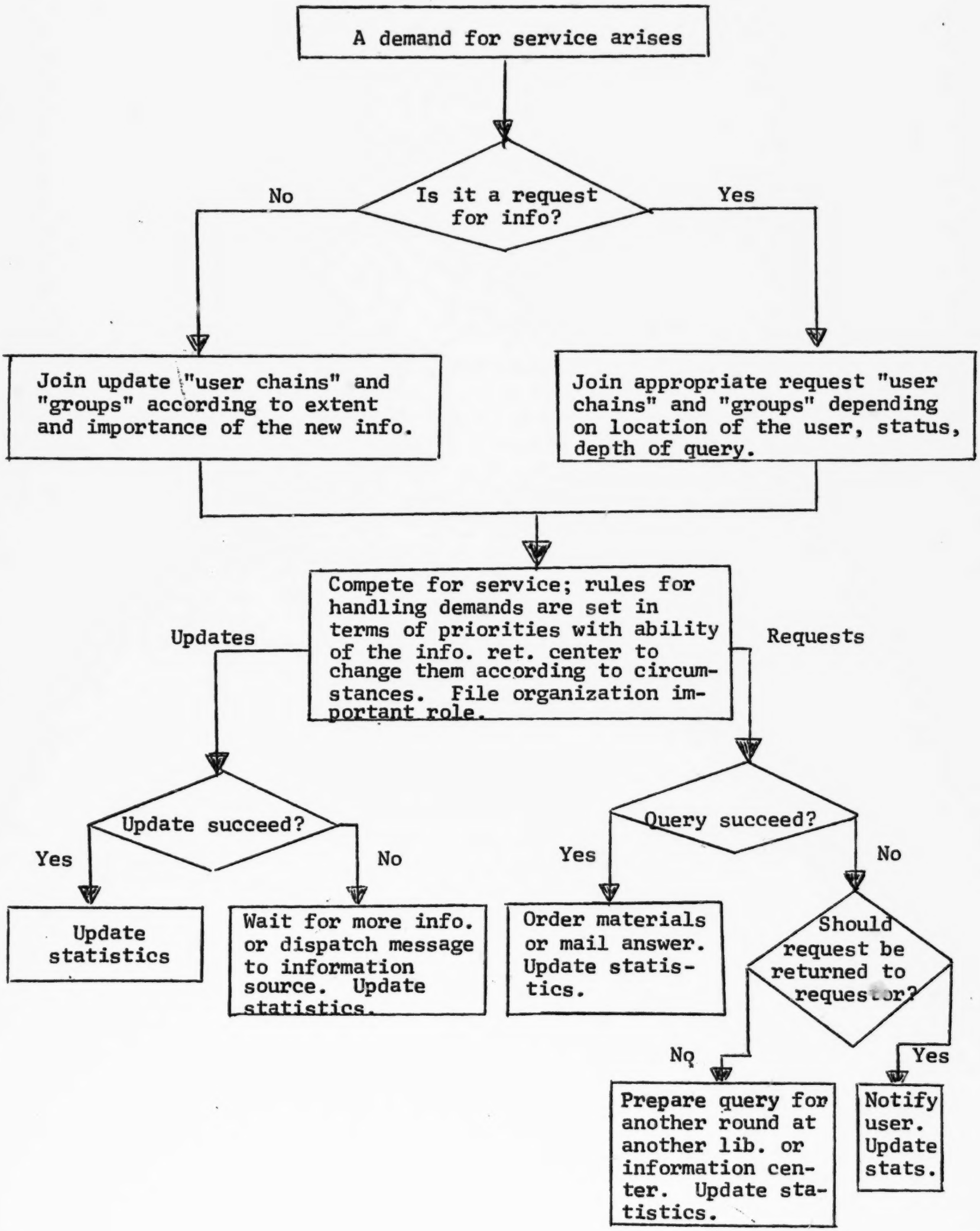

A demand for service arises
Is it a request for info?
No
Yes
Join update "user chains" and "groups" according to extent and importance of the new info.
Join appropriate request "user chains" and "groups" depending on location of the user, status, depth of query.
Compete for service; rules for handling demands are set in terms of priorities with ability of the info. ret. center to change them according to circumstances. File organization important role.
Updates
Requests
Update succeed?
Yes
No
Update statistics
Wait for more info. or dispatch message to information source. Update statistics.
Query succeed?
Yes
No
Order materials or mail answer. Update statistics.
Should request be returned to requestor?
No
Yes
Prepare query for another round at another lib. or information center. Update statistics.
Notify user. Update stats.

Figure 5: Gross level chart of model for file organization and update subsystem. Files are activity organized.

ORDER-AND-DELIVERY SYSTEM MODEL

A model that we discuss very quickly is the ordering-and-delivery system (see Figure 6). This system might also be called the conventional-materials distribution system since it is concerned to a very large extent with just that. Near completion is a comprehensive four-library version representing a bimodal network (e.g., the University of California). Some of its interesting features are:

- notion of partial satisfaction of a need.
- the utilization of the user-model (to be discussed next) in a mode appropriate to partial satisfaction and as an integral part of the model.
- input for computer processing center model of the type discussed earlier.
- the representation of library holdings in a very detailed manner using matrix savevalues for probabilities that a material is presently available.
- a four-level description of request types (fact, one material item, a survey, and exhaustive search) analogous to Goodman, et. al. (loc. cit.).

A nine-library version of this model is also being considered. It would allow a less schematic representation of the University of California library.

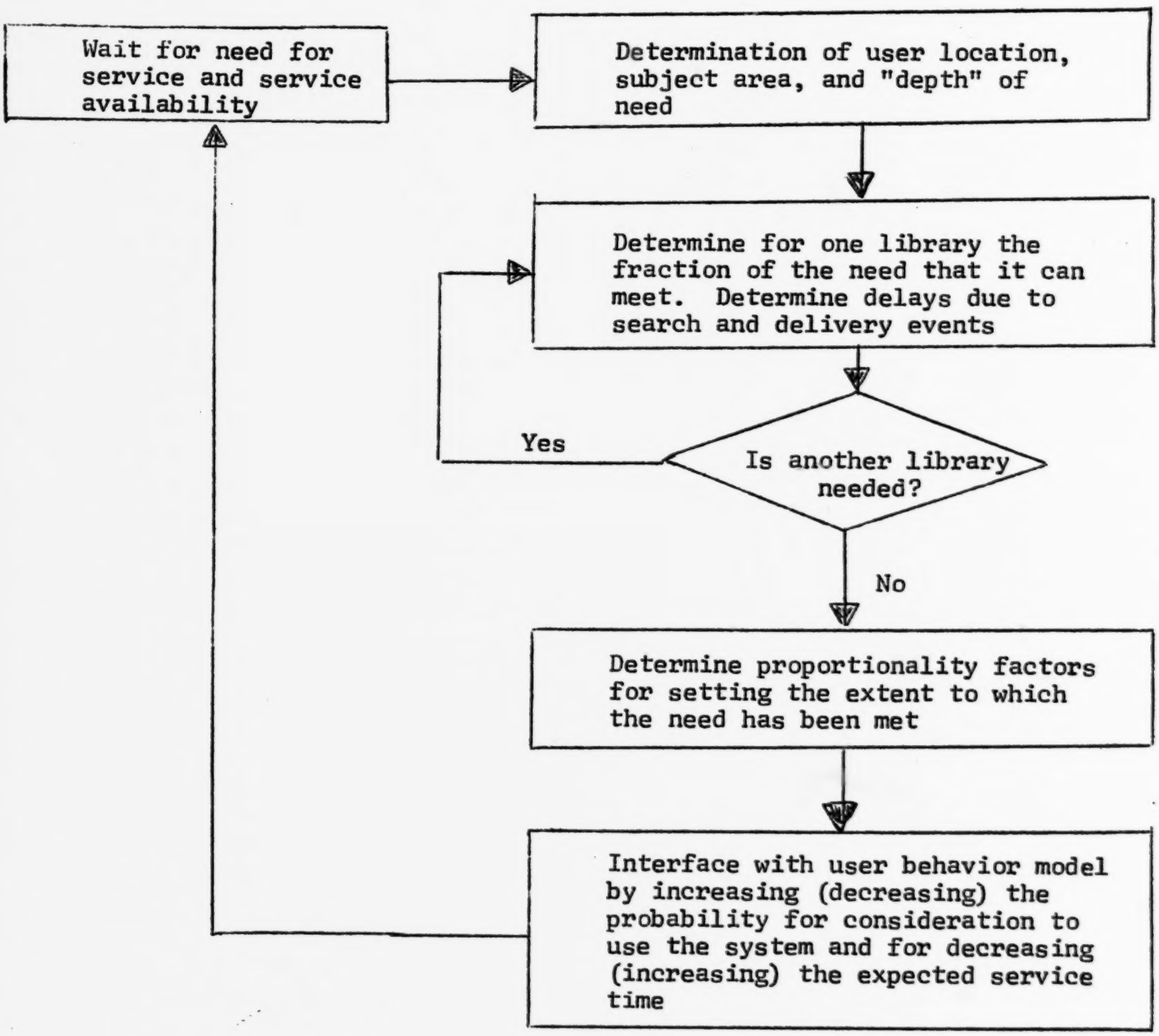

Wait for need for service and service availability
Determination of user location, subject area, and "depth" of need
Determine for one library the fraction of the need that it can meet. Determine delays due to search and delivery events
Is another library needed?
Yes
No
Determine proportionality factors for setting the extent to which the need has been met
Interface with user behavior model by increasing (decreasing) the probability for consideration to use the system and for decreasing (increasing) the expected service time

Figure 6: Gross level chart of model for taking account of distribution of materials in the network and for fractional completion of request requirements. Each library in the network is represented by an array of probabilities which give the probability of that library's meeting a request in a given subject area. An array of this type exists for four depths of service. The subject area of the request is determined and by invocation of a search strategy the path for satisfying the request is traced from library to library with timing of delivery of materials and search affecting actual service time.

USER-BEHAVIOR MODEL

In this section we discuss a (limited) version of the user-behavior model. One limitation that we impose (for discussion purposes) is that we are concerned with only one type of service, e.g., books.

Formulation of the Model

We first postulate the existence of a need time (NT) for our users, and a (actual) service time (AST) for our (one) material; these are both distributed quantities. We may also postulate that if, for any request,

$$AST \leq NT$$

at least one impediment to successful service has been removed. A second ingredient to success is convenience; if we get what we want when we want it only at the price of great effort we are not going to be happy. For simplicity, let the condition that $AST \leq NT$ be the only condition that determines user satisfaction.

Let us assume a mechanism of the following type for our user. Let him be partly "irrational": if he gets poor service on an occasion he is less likely to consider the library as a good place to go the next time he needs information. Let this emotion be dubbed the consideration probability (a threshold in the model), and be denoted by CON.

Let our user also be partly "rational". Let him retain an estimate of service times on the basis of his experiences (or on someone else's authority). Let this record consist of a parameter, the estimated service time, EST, interpreted as a threshold or as the mean of distribution whose dispersion he knows.

It should be clear from the above statements that we expect both CON and EST to change with experience (i.e., with each interaction with the library). Assume one need per one time unit and as a starting point, the following change rules:

$$CON_{t+1} = \begin{cases} CON_t & \text{if } E_o \\ (1-\theta)\, CON_t + \theta & \text{if } E_S \\ (1-\theta)\, CON_t & \text{if } E_F \end{cases}$$

$$EST_{t+1} = \begin{cases} EST_t & \text{if } E_o \\ A\ EST_t + B\ AST_t & \text{if } E_S \\ C\ EST_t + D\ AST_t & \text{if } E_F, \end{cases}$$

where the event E_o is a no-request event, E_S is a successful request, and E_F is a failed request; θ is a constant ($0 \leq \theta \leq 1$); A, B, C, D are constant, not restricted in value. The CON change rule is exactly the same as that in the Linear Model of Mathematical Learning Theory (see Hilgard and Bower, 1966). A high value of θ produces a fairly stable set of values for CON; near-unity values of A and B (e.g., .9) coupled with near-zero values of B and D (e.g., .1) produce a fairly stable set of EST values.

Taken together the statements of the previous paragraphs provide us with the user-behavior model presented in Figure 7. One other element of the model (not depicted in Figure 9) is the matter of costs and returns for various information policies that the user can invoke. We shall discuss this more below.

Elementary Model Results

Let us discuss some of the elementary results obtained from running a model of this type. Specific values for parameters must be assumed. (Relative values are of most importance in a comparative analysis.) For θ, we use the value .1; different values of A, B, C, D were used as indicated

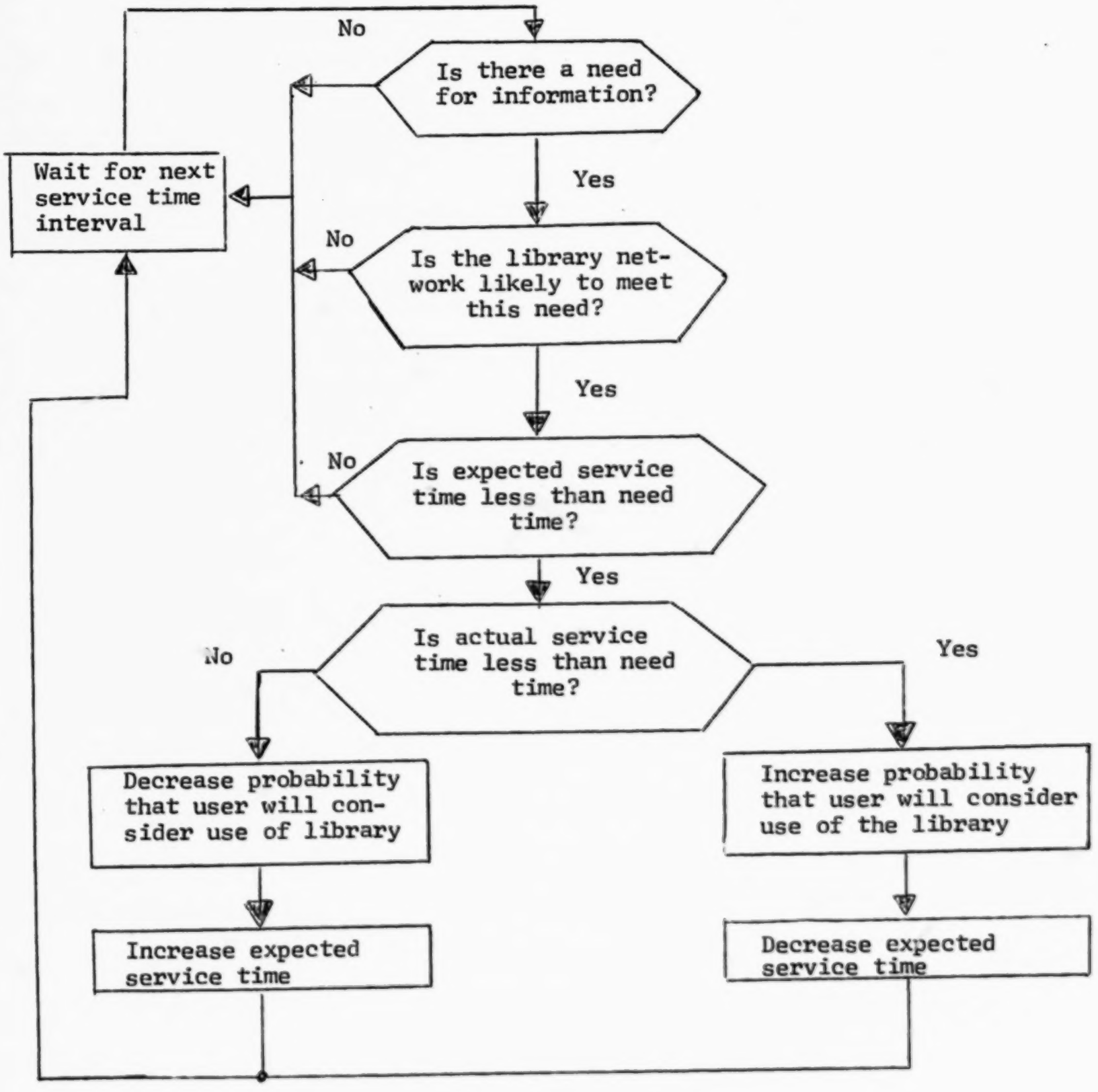

No
Is there a need for information?
Yes
Wait for next service time interval
No
Is the library network likely to meet this need?
Yes
No
Is expected service time less than need time?
Yes
Is actual service time less than need time?
No
Yes
Decrease probability that user will consider use of library
Increase probability that user will consider use of the library
Increase expected service time
Decrease expected service time

Figure 7: Gross level chart of user-behavior sybsystem model (simplified).

below. Initial values for CON and EST must also be assumed; a range of values of these were also utilized.

Figure 8 shows how the final value of the mean of CON is independent of the initial value--a fact that can be shown mathematically. A similar phenomenon obtains for EST.

Figure 9 illustrates the distribution of AST along with two EST distributions for cases when A = C = .9 and B = D = .1 and when A = B = C = D = 5, respectively. It can be seen that the EST distribution is confined within that for AST. This same result can be exhibited in the interval-by-interval plot of the values of EST and AST. This is due (quite clearly) to the fact that EST_{t+1} is, with the particular rules we have chosen here, a weighted average of the new AST value with the old EST and thus must always result in a value between these two quantities.

Evaluation Procedures

One of the purposes of estimating service times is to avoid making requests when there is a low probability of success. The effects of various policies can be seen in terms of a plot (or table) of values for needs, considerations, requests, and successes. Such a curve is presented in Figure 10. Figure 11 provides a rundown on these items in several cases:

1) CON mechanism only;

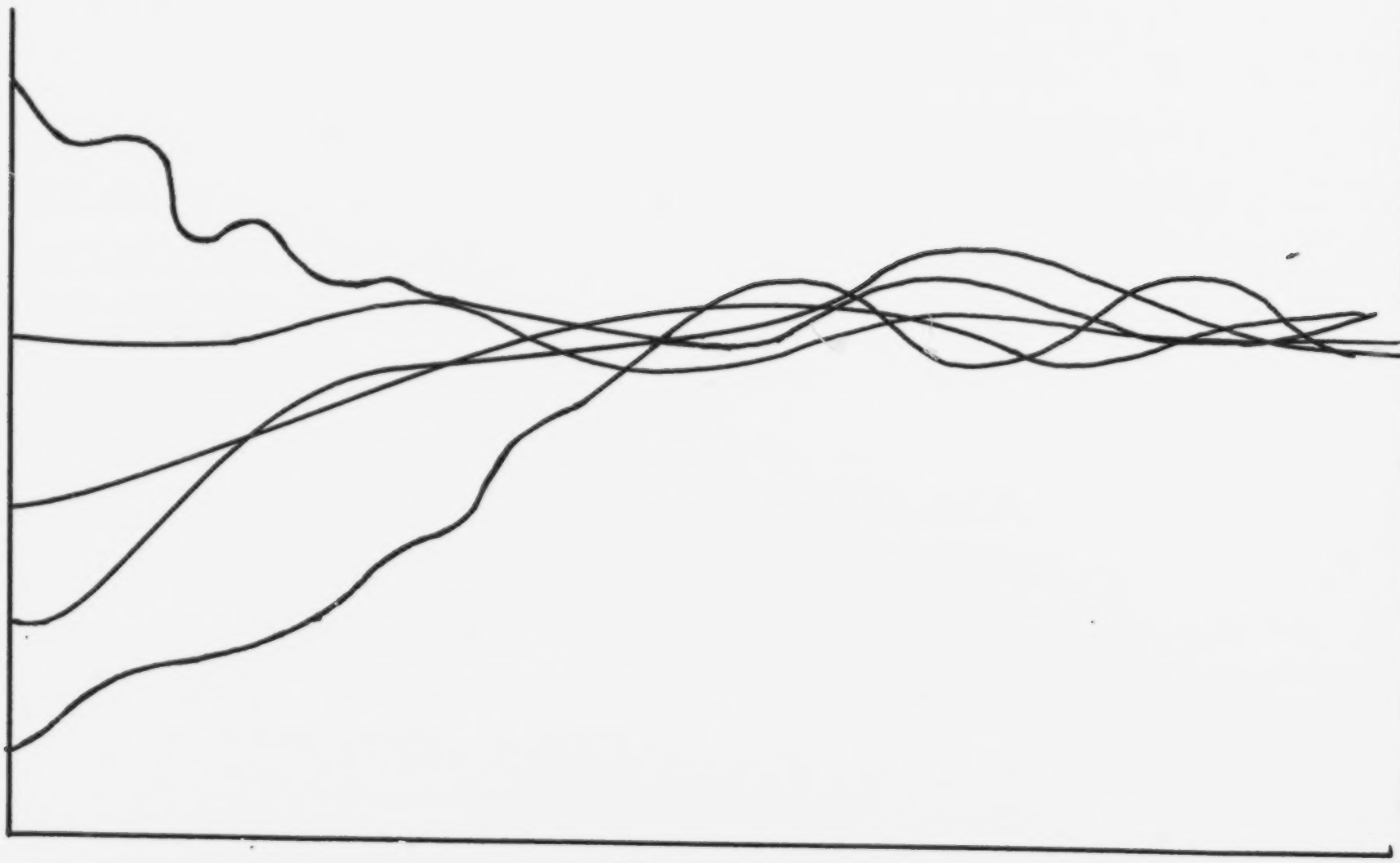
2) EST mechanism only; user estimates EST's mean; a distribution about the mean is provided and used in a purely guessing fashion.

3) CON and EST, the latter used as in the previous case;

4) CON and EST, the latter sectioned with user knowing when to expect long and short service times.

Evaluation in terms or costs and returns is useful. The costs for fulfilling requests should be relatively easy to represent in the model since they are related to system utilization. Returns, seemingly, are

PROBABILITY OF CONSIDERATION



TIME

Figure 8 : Probability of consideration to use the library network (averaged over types of needs as well as over the users) as a function of time, with different initial values.

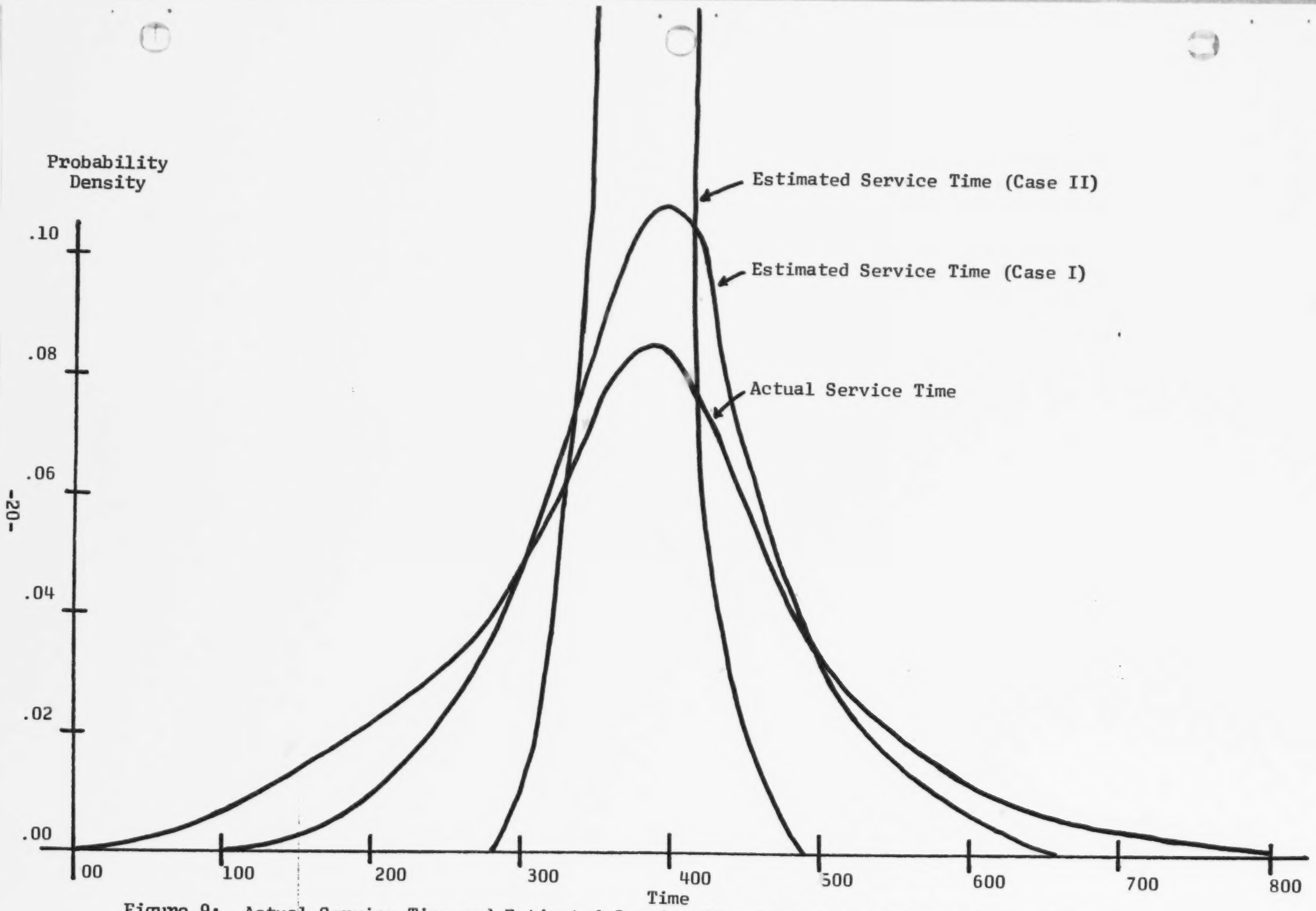


Figure 9: Actual Service Time and Estimated Service Time Distributions for a large number of trials. Case I refers to a 50-50 and Case II to a 90-10 weighting of the previous estimated service time and the actual service time (current) on each trial.

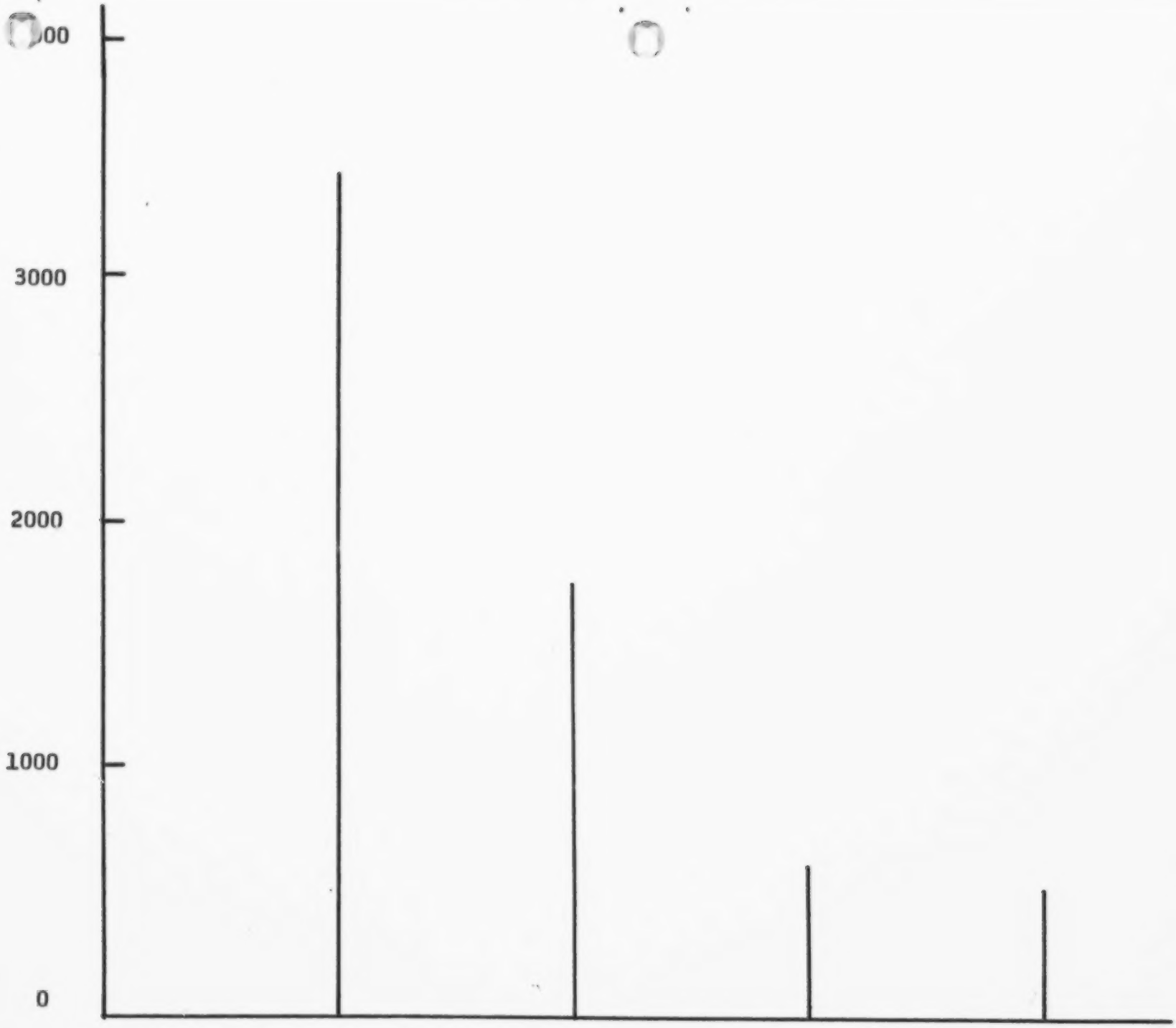


Figure 10: Number of needs, considerations to use the library, requests and responses meeting the needs.

	CON ONLY	EST ONLY	CON/EST GUESS MODE	TWO "EXPLOITATION" CLASSES
NEEDS	1,000	1,000	1,000	1,000
CONS	560	560	730	843
REQS	560	560	410	528
SUCCESS	331	417	310	454

Figure 11: Needs, considerations, requests, and successes for four cases. (See text for fuller description of the cases.)

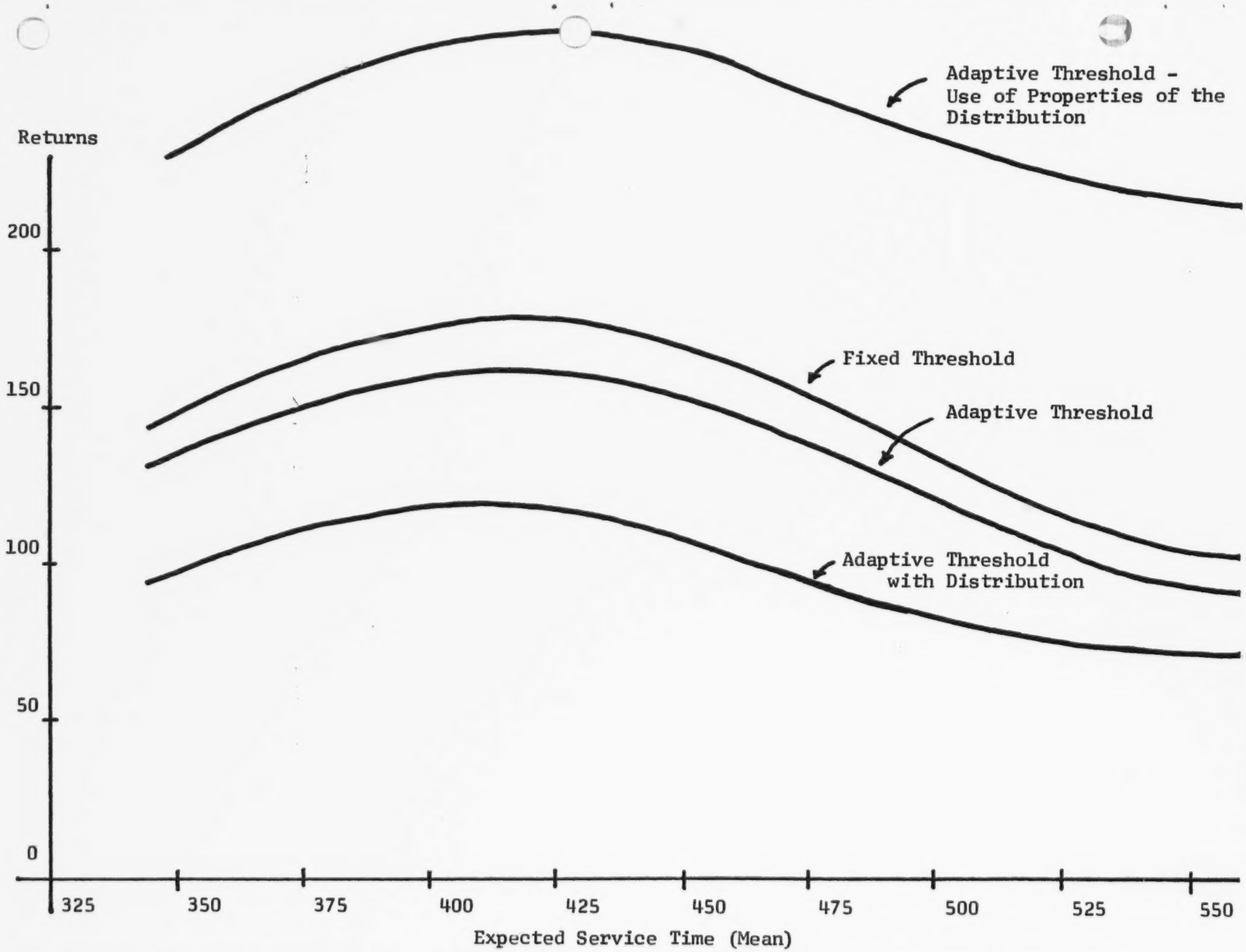


Figure 12: Net returns under different user strategies on assumption of a particular return rate. In those cases where EST is distributed the abscissa is a mean.

not easy to express. The exact treatment of them is not, however, the issue here. Our concern is this: "Given a cost/return paradigm, how valuable is it for the user to be able to accurately predict the AST?" For illustrative purposes only, we assumed a fixed cost (e.g., a mean value) for each request and a fixed return for each successful request. The values that we used implied a fairly high overall returns rate (20-40%). A high rate might be defended by analogy to that associated with obtaining or extending education. The results of our analysis are illustrated graphically in Figure 14 for the following four cases:

1) EST used in a guessing mode, as described above;
2) EST as a threshold (no distribution);
3) a fixed threshold (with various assumed values);
4) EST with expectations of long and short service times.

GPSS AND STATISTICS

We have already seen a bird's-eye view of the statistics provided automatically by the GPSS program. It just happens from the nature of many models (e.g., our Blunt-like model) a great deal of mileage can be had without recourse to further statistical effort. In other cases correlations between variables, etc. are essential to the expression of model results.

The quickest path to more statistical power in GPSS is to link it to the FORTRAN libraries of statistical routing (e.g., the Scientific Subroutine Package (SSP) of IBM, the UCLA Biomedical statistical programs (BMD), etc.). In order to make the link successfully it is necessary to do a certain amount of programming in IBM 360 Assembly Language in context of the GPSS HELP block. Let us briefly discuss how this might be done for a particular example. The program that we wish to link to is a FORTRAN chi-square program. The data to be analysed is contained in the parameters of the "trigger" transaction (the one that goes through the HELP block).

We begin with the GPSS program in which the following statement is encountered:

```
HELP     EFCHI
```

This brings into core a load module consisting of an Assembly Language Interface and the (compiled) FORTRAN chi-square subroutine and any routines it may call. Control transfers to this program.

The first stage in the Assembly interface is spent in going after the parameters of the "trigger" transaction. (GPSS provides a set of control words, located through Register 10, which allow us to get at these data.) Before we can do anything with any parameters we must first take into account whether they are half- or full-words. Having done that we can move out the first two numbers giving the size of the data array (N, the number of rows and M, the number of columns). We also at this stage move X'48' or X'58' into a location symbolically addressed by the name OPCODE. OPCODE sits in the middle of a line of code at the very point a L or LH is required (inside a loop) to move individually each parameter out of the GPSS storage location into a register. With the parameter value in this register a branch is then made to a GPSS-provided routine (FLOAT) for converting it from fixed point to floating point. Upon return from this routine the converted parameter value is stored in an area previously set aside for it in another (GPSS) routine, GETCOR. The address of the area where the converted parameter ends up was set, upon return from GETCOR in AAD. AAD is one member of the "parameter list" (see Figure 13) required by the FORTRAN chi-square program. At the end of the data conversion stage control passes to chi-square. Finally, chi-square's output data must be reconverted and stored in the parameters of the "trigger" transaction (or in savevalues) through a series of statements that reverse those just outlined.

The above example describes the situation in a particular case. An

```
PARAM       EQU   *
*           INPUT
AAD         DS    F address of array of chi square input data
            DC    A(N) address of N the number of rows
            DC    A(M) address of M, the number of columns
*           OUTPUT
CSAD        DC    A(CS) address for CS, chi-square
NDFAD       DC    A(NDF) address of NDF, the number of degress of freedom
            DC    A(IERR) address of error code
*           WORK
TRAD        DS    F   set in program
TCAD        DS    F   set in program
```

Figure 13: Parameter list for FORTRAN chi-square routine.

interesting problem arises if we wish to pass other GPSS data types (e.g., parameters, savevalues, matrix savevalues, block counts, entity attributes) to the assembly program. (It may be feasible to short circuit this need for generality by passing all data to parameters and savevalues in the GPSS program before passing into the Assembler phase; however, this may not always be possible.) A solution to this problem seems possible through introduction of general HELP block, STAT , the fields of which contain proper control information for linking to any data type. It is also necessary to have information in either these fields or elsewhere to denote the statistical routine that is required. The "elsewhere" most likely is the parameters of the triggering transaction. In order to utilize such a system it would be necessary to develop a user's manual (similar, in some ways, to those developed for BMD and SSP) describing how to set up the fields of this block and what numbers to assign to the parameters of the incoming "trigger" transaction. An example, in which the origination of the data, the desired routine, and the size of the input arrays are set up in the fields of STAT is the following:

```
HELP        STAT, 1101, 1001, 1901, 3, 3
```

where: A = STAT = name of the HELP routine

B = 1101 = chi-square test (non-parametric tests = 1100)

C = 1001 = input data is in savevalues (1000); first data item is in X1

D = 1901 = output data is to be returned to savevalues (1000) starting at X901.

E = 3 = number of rows in input table

F = 3 = number of columns in input table.

The Assembly code would have to be supplied with information to decode these (coded) values and to set switches as to where to branch for calling the proper subroutine.

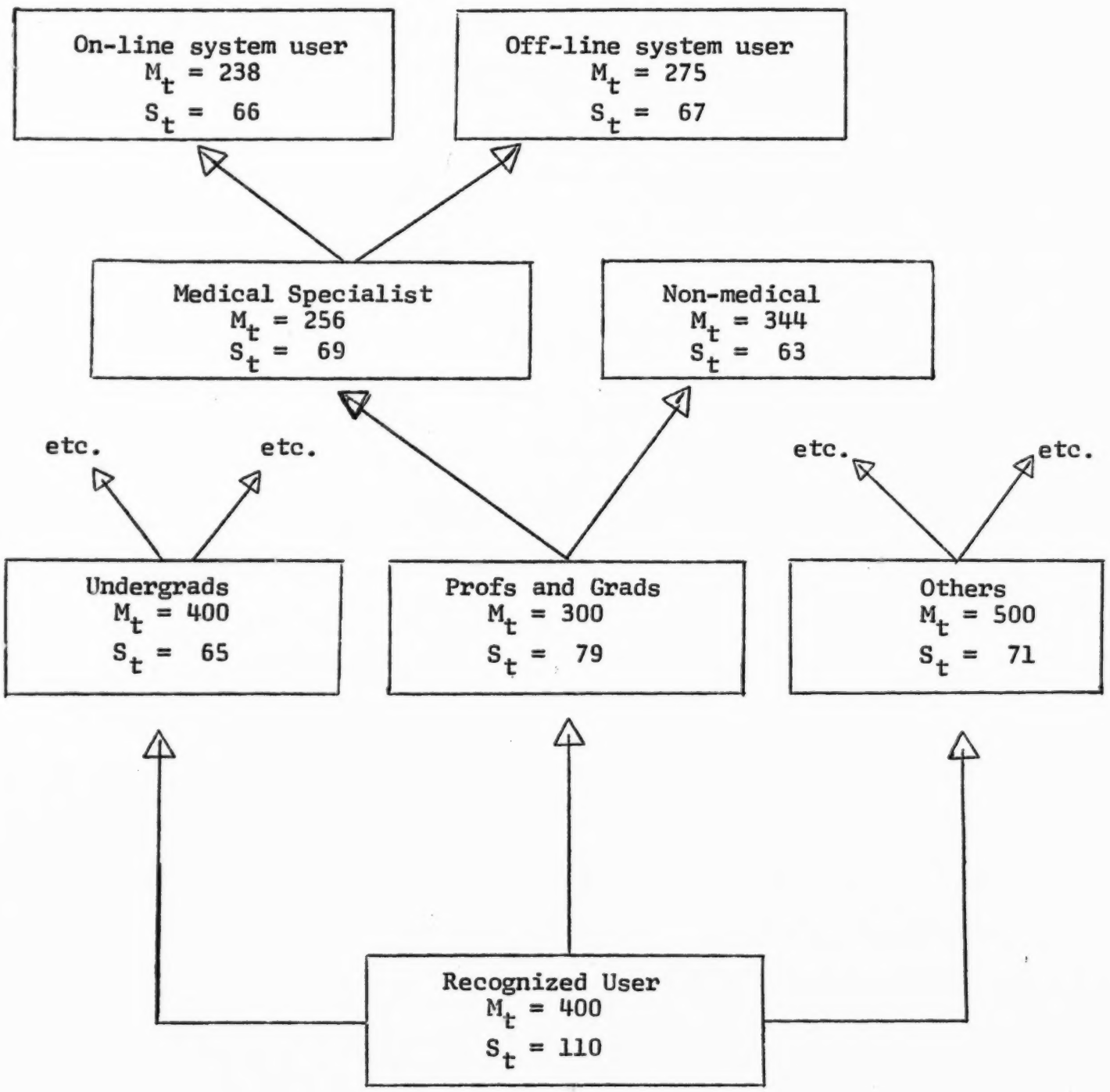


Figure 14: Example of prediction tree, depicting service time means and standard deviations as a function of information known about the user.

In passing we would like to note that linking to PL/1 seems desirable. A program for manipulation of prediction trees is possible utilizing the list-processing capabilities of PL/1. An example of the kind of output to be realized in such a case is that shown in Figure 14, wherein the response time mean and standard deviation are predicted according to amount of information we have on the user. Many of the basic PL/1 capabilities for manipulating such trees have been investigated as has the linkage from GPSS. We should soon have a working version of such a program.

REFERENCES

Baker, N. R. and R. E. Nance. "The Use of Simulation in Studying Information Storage and Retrieval Systems." American Documentation. November, 1968.

Blunt, C., Duquet, R. T., and P. T. Lucki. A General Model for Simulating Information Storage and Retrieval Systems. HRB-Singer, Inc. Science Park, State College, Pennsylvania, April, 1966.

Bourne, C. P. and D. F. Ford. "Cost Analysis and Simulation Procedures for Evaluation of Large Information Systems." Amer. Doc., 15, 142-149, April, 1964.

Brown, G. W., Miller, James G., and T. A. Keenan. EDUNET: Report of the Summer Study on Information Networks. New York: J. Wiley, 1967.

Goodman, A. F., J. D. Hodges, Jr., et. al. Final Report, DOD User-Needs Study, Phase II: Flow of Scientific and Technical Information within the Defense Industry. Autonetics Division Report C6-2442/030, Volumes I, II, and III (AD 647 111, AD 647 112, and AD 649 284), North American Rockwell Corporation, November, 1966.

Goodman, A. F. and S. O. Jones. User Information Needs: The Challenge and a Response. A paper presented at the 31st Meeting of the American Society for Information Science, October 20-24, 1968. (Douglas Paper 10,000, Mc Donnell-Douglas Astronautics Co.)

Hilgard, E. and G. Bower. Theories of Learning. 3rd Edition. New York: Appleton-Century-Crofts, 1966.

Jones, M. M. Incremental Simulation on a Time-Sharing Computer. Cambridge, Mass.: Project MAC, MIT., 1968.

Reilly, K. "Outline for a Simulation Study of the California State Library Network." Part 5 of the Final Report on Specification of a Mechanized Center for Information Services for a Public Library Reference Center. Los Angeles, California: University of California Institute of Library Research, July 1, 1968. (1968a)

Reilly, K. D. "Digital Computer Simulation Studies of Information Networks." In Digest of the Second Conference on the Applications of Simulation, IEEE Catalog No. 68C60-SIM. (1968b)

Reilly, K. D. and R. M. Hayes. The Effect of Response Time Upon Utilization of an Information Storage and Retrieval System--A Simulation. A paper delivered at the Annual Meeting of the Operations Research Society of America, June 1-2, 1967. Available as a report of the Institute of Library Research, University of California, Los Angeles, California.